BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

AppShield™ 4.0

White Paper

Sanctum, Inc.

October, 2003

Table of Contents

1 Executive Summary

E-Business has transformed the landscape in which applications are used—from an environment of limited access to one providing wide open, "24x7" admission. This has created new and more difficult security problems that many companies are only beginning to discover.

Today, Internet security is comprised of four elements: 1) antivirus protection at the desktop, 2) data encryption and authentication for transport, 3) network firewalls and advanced routers at the network-layer security, and 4) manual patching for application-layer security. Encryption and virtual private networks, using algorithms such as SSL, provide security for data traveling over the public Internet. Firewalls prevent unauthorized network-level access to the server systems on which e-Business applications reside. The reality is, neither network firewalls nor encryption schemes like SSL protect the web application itself. Web application security ensures that Web applications can only be used the way they were intended by the developer. Any attempt at manipulating them is directly blocked, preventing the unauthorized use of an e-Business' resources or customer information by hackers attempting to gain access to the online network directly through the application itself.

When the web was developed, the original concept of a web application did not exist. In fact, the web's original developers never thought of the web as more than an effective method to deliver static content that would be updated and published much like a book. Quickly, functionality was added to collect input from the user, but the concept of a web application would not fully develop until the web servers were connected to the data bases themselves and web pages were no longer written by hand but generated and customized based on a users request. Today, Web applications are comprised of Web servers, User interface code, front and back end applications, and databases. Thus Web applications today house the most valuable assets a company has, namely their digital assets and data.

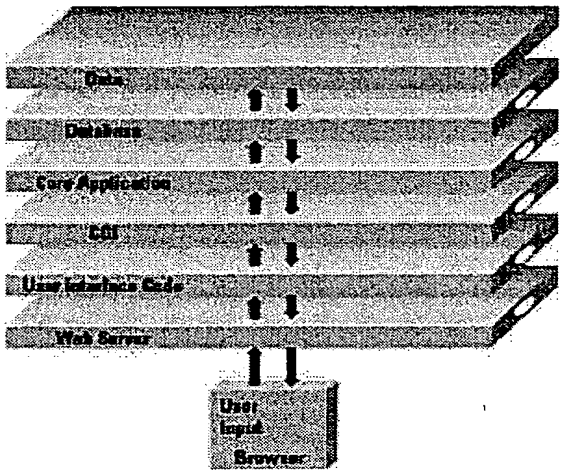

Figure 1: Anatomy of a Web application

Current approaches to web application security address security issues at the last and most expensive stage of the development cycle—deployment. This approach often includes a line-by-line code review for security holes, sometimes at a cost of up to 50 cents per line. In fact, most sites add so much new code every day that they could never hope to keep up by patching or fixing holes manually, making the majority of sites insecure. Because security is checked at this late stage, and a never-ending stream of security patches are released by 3rd party vendors, the process becomes one of great expense, consuming time and resources through the required vigil the site owner must hold watching for potential security vulnerabilities. This rapid charge of Web application development has made "24/7" uptime and site security impossible.

To address these issues, Sanctum has developed AppShield™ software, the industry's leading Web Application Firewall solution. Sanctum's AppShield software is a Web Application Firewall built on a security proxy architecture that enforces a positive security model blocking any type of application manipulation. It is an active system that monitors and responds to any unusual or unauthorized behavior anywhere within a site, blocking attacks before they can reach the site.

The web application security provided by AppShield ensures that Web applications (both the application and Web logic) can only be used the way they were intended by the developer. Any attempt at manipulating them is directly blocked, preventing the unauthorized use of an e-Business' resources or customer information. Even un-patched and insecure systems are made secure by AppShield allowing site owners to perform security patching with regularly scheduled maintenance. Most importantly, **AppShield does not require any patches or signature files to stay up to date,** so the unpredictable and costly cycle of security vulnerabilities and patching is broken once and for all.

AppShield Provides Enterprise Strength Application Security

- ➔ Prevents application level attacks such as:
    1. Stealing company assets such as employee files, will be prohibited
    2. Commerce sites will be protected against anyone falsifying buy/sell transactions
    3. Hijacking confidential customer information, such as financial portfolios, will not be allowed
    4. Site Defacement will be impossible
- ➔ Protects company assets, reputation, and revenue streams
- ➔ Assures consumers that their privacy is protected

AppShield is automatic

- ➔ Automatically secures existing and future applications
- ➔ Automatically secures third-party and custom applications

AppShield is designed for e-Business

- ➔ Allows for rapid application development and deployment
- ➔ Enables greater focus on core competencies
- ➔ Is tuned for high performance, scalability, availability, and manageability

AppShield offloads the application security responsibilities from the development team by providing an automatic solution to a very complex and daunting problem. AppShield is the ideal solution for any organization, from the large multimillion pages-per-day operation, to the small single-server site. *When you are Secured by Sanctum, you can be sure your valuable digital assets are secure.*

2 Overview

AppShield enforces a **positive security model**. AppShield sits as a proxy between the network firewall and the web server. AppShield employs Sanctum's unique, patented **Dynamic Policy Recognition Engine (DPRE)** technology to examine and enforce application security policies in real time with the most powerful HTTP security proxy ever developed. This type of positive policy requires no negative signature database, is unique to each user session, is always accurate and up to date, and requires minimal administration.

In addition to the DPRE, AppShield provides a powerful automated policy management system allowing the site owner to customize and control the site policy easily and quickly. AppShield easily and securely supports even the most complicated and dynamic web sites using client side logic, such as Java, JavaScript, Flash or ActiveX components.

Built on a highly scalable proxy platform, AppShield can support sites receiving millions of hits per day. Multiple AppShields can be linked together providing even greater scalability as well as a truly fault tolerant solution.

2.1 The Positive Security Model

A positive security model enforces intended behavior vs. watching for unintended behavior. In other words, positive security only permits good behavior vs. preventing bad behavior. Positive security assumes an administrator and/or developer can define the ways in which you want a user to interact with your site, compared to the virtual impossibility of defining all the ways in which you think someone may try to manipulate/hack/misuse your site.

The benefits of a positive security model are:

1. Positive security policies do not require patches, signatures, or continual updates. And they protect against unknown vulnerabilities.

2. A positive security model contains a complete set of valid requests. There are no unknowns. Thus, the number of false negative and positives is significantly reduced.

3. Positive policies have a better ROI. They require little main memory and no disk space making them very efficient at processing requests.

4. Positive Security helps to significantly lower Operating Costs due to less administrative overhead since no continual updating is required, and the elimination of unplanned maintenance downtime

2.1.1 Security Model Comparison Table

Positive Security Model	Negative Security Model
Complete	Incomplete
Accurate	Uncertain
Efficient	Wasteful
Non-signature based	Signature based
Low Admin	Ongoing Admin
Small Footprint	Large Footprint
Low Resource Usage	Heavy Resource Usage
Non-disruptive	Disruptive
No unknown requests	Unknown requests (good & bad)

2.2 Protecting the Application as well as the Web Server

Your web application is unique. It is developed for the needs of your customers, partners or employees. Any vulnerability found in your application will not be widely known, cannot be patched by anyone but you, and will most likely go undetected until it's too late.

Protecting against application vulnerabilities is much more difficult and more important than protecting only the web server itself. AppShield elegantly solves this problem by using Sanctum's patented Dynamic Policy Recognition Engine (DPRE). The DPRE analyzes all data passed between the client and the server, closely watching all of the requests that are made by a user, and making sure that the values sent in the URL, cookies, hidden fields or any other HTTP elements are not modified by the user. AppShield is aware of all the links and possible choices a user has at any time during their interaction with your web site and restricts their actions to only the choices they have been presented. This enforcement places a sandbox around every user visiting your site, effectively walling them off from other users, and forcing them to follow the intended browsing path your site was designed with.

In addition to protecting the application logic, AppShield's DPRE also protects the web server against all known and unknown attacks **without signatures or updates**. This includes known vulnerabilities, server-based worms, holes or mis-configurations in the web server itself, as well as new attacks for which no patch exists. Because AppShield's DPRE can effectively identify the valid requests, unknown attacks are easily blocked.

AppShield's Dynamic Policy Recognition Engine automatically identifies the security policy of each HTML page by processing in real-time the page elements (Figure 2)

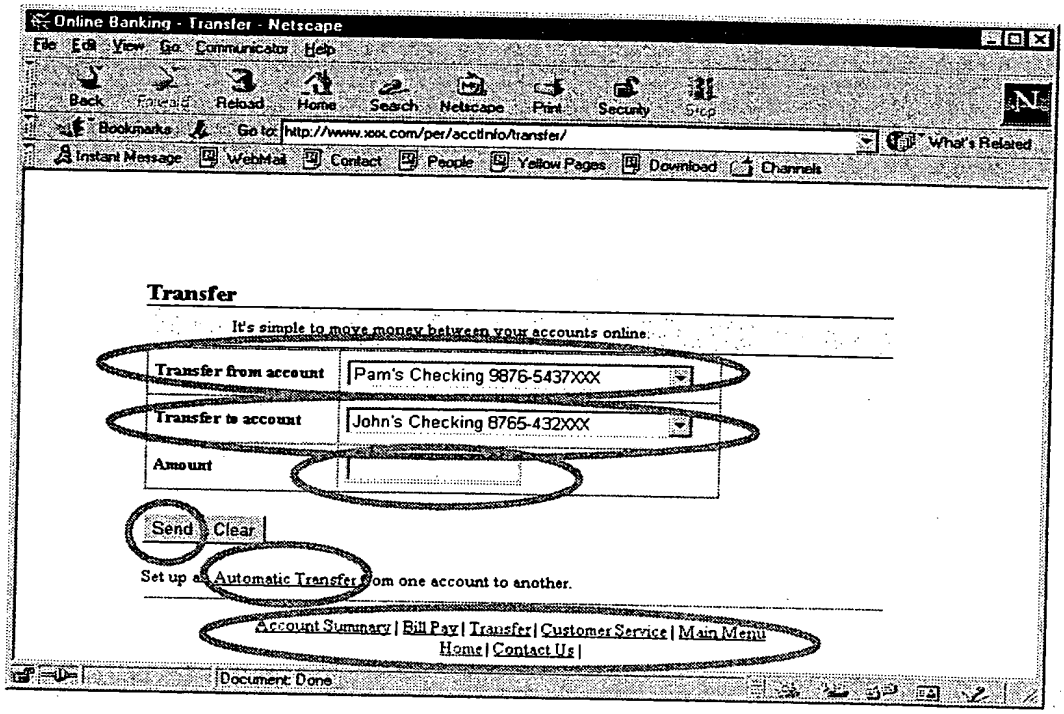

Figure 2

In the case of a hacking attempt, the Dynamic Policy Recognition Engine will determine an illegal request is being sent to the application. Instead of relaying the illegal request, AppShield logs and blocks the request and then invokes a *Response Page* (Figure 3) which dynamically generates a customized message that is sent to the user informing them of their actions and explaining what they might have done wrong without terminating the user session. If a user continues submitting illegal requests beyond the configured threshold settings, then AppShield will terminate the offending user session. AppShield also invokes a *timeout page* in case a request is sent after a users session has timed-out.

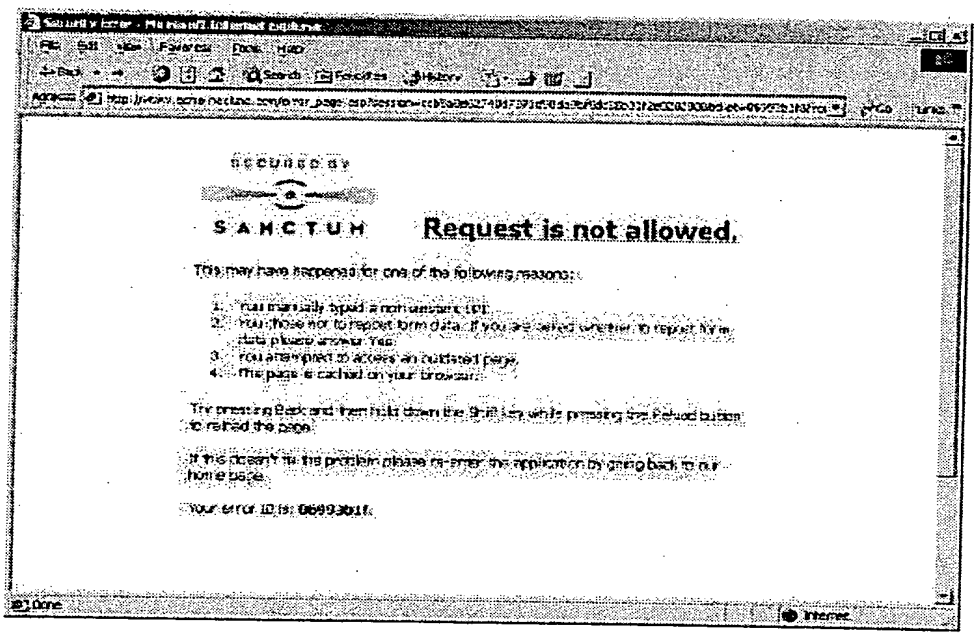

Figure 3

2.3 The Client Side: Trusting the Un-trusted

For a web application to keep track of its users, each user must "remind" the web application with every request who they are. Traditionally, this was done using cookies, which were designed for this task. The web application would set a unique value or session ID in a specific cookie and send that to the browser after the user had logged in. Automatically from that moment on, the web browser would send the cookie information with every request. Several other methods have been offered for users not wishing to use cookies, however, they all require that some form of session identifier be stored on the client and sent with every request.

The problem with these methods is that **anything stored on the client, can be modified by the client**. For example the following URL describes a site where the session ID is stored in the URL:

HTTP://www.xyz.com/index.html?sessionID=5

Using only a web browser, one can go from user to hacker in seconds by simply modifying the URL itself. Where I had a **sessionID=5** in the valid request, I can easily change this to **sessionID=7** and enter or access the information for another user, without any authentication.

To combat this simple problem, SSL ID matching, IP matching, and session ID encryption have been added to the mix, all significantly increasing the complexity of the web application. None of these solutions can guarantee or even notify a security administrator when a user is attempting to modify any

value on the client side. In the end, the web application is forced to trust the client side that extends far from the trusted environment of the web server passing through the Internet and onto the desktop of any person in the world with a web browser.

AppShield immediately detects and prevents application hacking attempts before they happen, such as:

1. **Hidden Manipulation** - Modification of state information stored inside hidden fields
2. **Buffer Overflow Attacks** - Sending too much data in a request to the application, attacking either 3rd party or internally developed code.
3. **Parameter Tampering** - Manipulation of the parameters being passes in the HTML (changing content of parameters, adding parameters, deleting parameters, etc.)
4. **Known Vulnerability Protection** – Problems in commercial products (web servers, application servers, etc.)
5. **Cross Site Scripting** - Inserting scripting languages into text fields to be displayed to other users
6. **Forceful Browsing** - Jumping directly to pages that can normally only be accessed through authentication mechanisms
7. **Stealth Commanding** - Planting Trojan horses in text fields that cause the web application to perform commands it is not intended to do
8. **Backdoor and Debug Options** - Exploiting vulnerabilities left open in internally developed code
9. **3rd Party Misconfiguration** - Exploiting configuration errors in 3rd party components, such as web and database servers
10. **Cookie Poisoning** - Changing a cookie's content

3 Architecture and Technology

3.1 AppShield Architecture

AppShield uses a secure proxy to provide the platform for AppShield's Dynamic Policy Recognition Engine. The benefits of proxies are well known and provide a true barrier between the outside world and your web applications. AppShield's secure proxy evaluates every request for RFC compliance, buffer overflow attacks, and invalid HTTP headers as well as translating all requests to a common format before passing the requests to the security engine. Because all requests must be understood and well formed during this process, all encoding type attacks will fail, as the true request will be revealed for what it really is.

When a user starts an application session by directing his browser to an e-Business site, AppShield first verifies that the page accessed is indeed a legal *Start URL*[1] for the site, a previously bookmarked page, or a signed link. For example, the site administrator may declare the home page to be a legal *Start URL* as well as any page under the "Products" section. After the initial check is done, AppShield creates an *application session token* and stores it inside a cookie[2]. This cookie is used in all future transactions to uniquely identify users. The AppShield cookie also has an added advantage in that it can be used to detect and prevent TCP/IP session hijacking (sequence number guessing) thereby maintaining and protecting authorized sessions.

Once a session is established, AppShield analyzes each HTML page that belongs to that session as it is being forwarded to the browser. The patented *Policy Recognition Engine* examines the page, looking for information such as CGI parameters, hidden field values, drop-down menu values, and maximum size of expected text fields. Based upon this run-time analysis, AppShield automatically determines the possible valid requests and builds a positive security policy for the application. As the web server generates more pages, AppShield dynamically/automatically adjusts the security policy for the session allowing for only valid requests.

In addition to securing the web server, AppShield can also protect the entire application, including backend data systems, application servers or legacy systems, as well as client code (client side logic or CSL) such as JavaScript, Java, ActiveX. It is important to understand that when using CSL such as a java component, this component is executing in an *un-trusted* environment and its behavior needs to be closely watched. Just as all user requests are validated against the AppShield dynamic policy, so are the requests generated by any client code. Because of the unique ability of AppShield to carefully control and monitor these requests, vulnerabilities that may exist in either your web application or in any client side logic are rendered harmless.

Extending the security to client side logic is done using the Intelligent Policy Creation and Editing tools. These mechanisms assist the user by automating the definition of policy rules that tell AppShield how to handle requests that are generated or modified by a program such as JavaScript on the client. For example, if the application employs JavaScript code that pre-loads or animates GIFs taken from directory /images/, AppShield generates a rule to allow all of the requests of the form /images/<legal_filename>.GIF. Other rules can be used to inform AppShield about operations such as hidden fields or cookies that may be manipulated by a programming language on the client.

---

[1] Defining the *Start URLs* is one of the few configuration activities required when installing *AppShield*.
[2] To prevent cookie poisoning, AppShield digitally signs the content of the original cookie and only then appends the application session token.

4 Installation and Configuration

Installing and configuring AppShield is a simple and understandable process. AppShield installation consists of two simple steps: Installing the AppShield security engine on each AppShield machine (or web server) and then installing the management console to complete the configuration process. The basic configuration required is simply to specify the IP of the AppShield system and the IP of the web server to be protected. Once this is configured, AppShield can begin to protect and process the traffic for your web application.

AppShield's network and system configuration tool, which is part of the management console, can be run remotely to configure all AppShield nodes simultaneously.

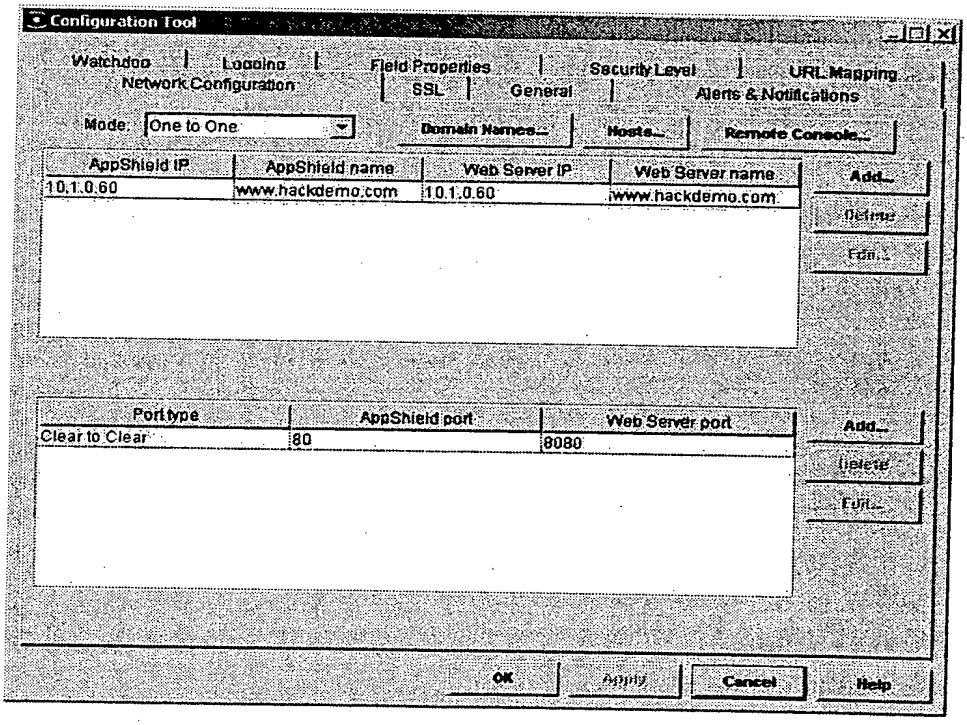

Figure 4

In most cases, very little security configuration is required since the Dynamic Policy Recognition Engine automatically enforces the most secure configuration for your site. If your site employs heavy use of client side code, AppShield has intelligent Policy generation tools to facilitate rapid deployment while still achieving a high level of security. As previously discussed, client side code poses extenuating security challenges in securing web applications. AppShield has a very simple and elegant way of creating policy such that the site, including the client side code, is protected. Sites using client side code such as Java, ActiveX, JavaScript and others can be quickly configured to work with AppShield without manually writing policy rules. This policy generation process is critical in ensuring the safe execution of client side code. Without this mechanism, the security of the web site cannot be ensured.

4.1 Policy Management System

The AppShield Policy Management System provides a dashboard from which the policy can be created and managed. The AppShield Security Templates provide a 'quick start' for implementing site-specific policy. While the Automated Policy Generating tool facilitates the rapid learning of your sites behavior and automatically generates policy that can be used instantaneously.

4.1.1 Security Templates

In the Security Level option, you define the Security Template you want AppShield to start from on your site. You may choose between three predefined security levels or customize your own unique level.

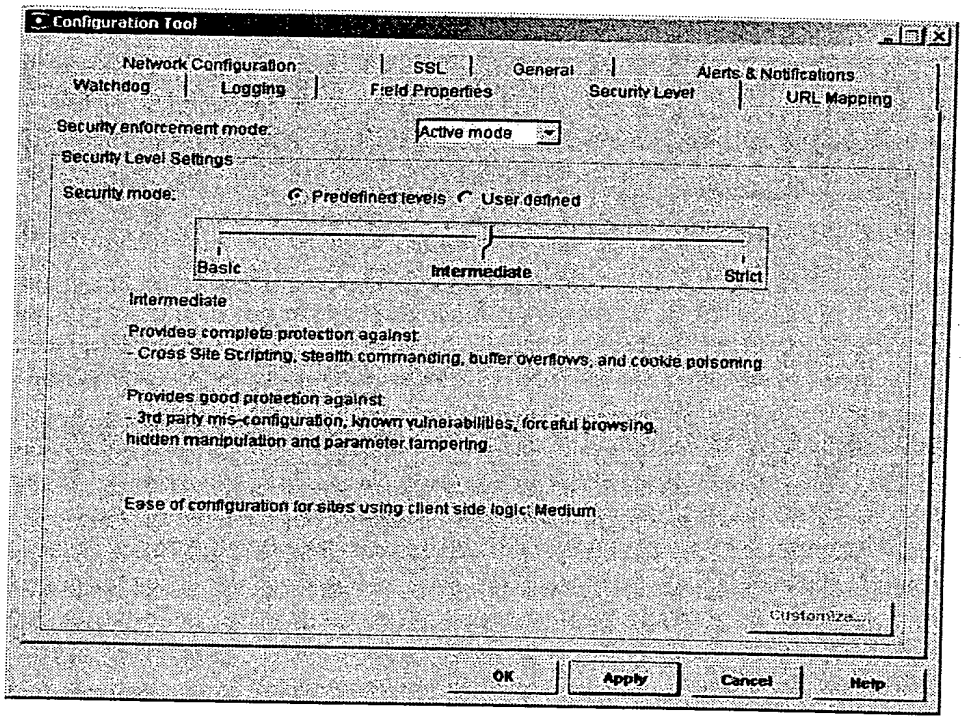

Figure 5

The predefined options are based on Sanctum's security recommendations to find the optimal balance between ease of configuration and level of security desired. The custom security level option allows the advanced user to determine their own security settings based on their intimate knowledge of the site and application security principles. There are three predefined Security levels.

Strict

Provides complete protection against: cross-site scripting, stealth commanding, buffer overflows, cookie poisoning, third party misconfiguration, known vulnerabilities, forceful browsing, hidden manipulation and parameter tempering. If the site contains a lot of CSL AppShield will require the use of the Policy Creation and Editing tools to tailor the policy specifically to the site.

Intermediate

Provides complete protection against: cross-site scripting, stealth commanding, buffer overflows, cookie poisoning. Provides good protection against: third party misconfiguration, known vulnerabilities, forceful browsing, hidden manipulation and parameter tempering. If the site contains a lot of CSL AppShield will require some use of the Policy Creation and Editing tools to tailor the policy specifically to the site.

Basic

Provides complete protection against: cross-site scripting, stealth commanding, buffer overflows, cookie poisoning. Provides good protection against: third party mis-configuration, known vulnerabilities, forceful browsing. Provides limited protection against: hidden manipulation and parameter tempering. If the site contains a lot of CSL AppShield will likely go in configured for the site and the Policy Creation and Editing tools may be used to tailor the policy specifically to the site.

Custom Security Levels

By using the user defined option the administrator can tune and configure the security of their site to the exact levels they desire.

4.1.2 Automatic Policy Generator

In AppShield, it is not necessary for the administrator to understand how to write policy rules or to understand how the application is intended to function. All that is required is a workstation with a web browser to automatically create the security policies needed. In AppShield you simply specify the IP address that is to be used as a trusted source, and all requests from that system will be used to create any rules required by AppShield.

4.1.3 Policy Organizer

The Policy Organizer's rule manager allows you to have a simple view of all the individual policy rules configured. You can sort them by any property and also define the "application" associated with each rule, so you can manage each group of rules separately.

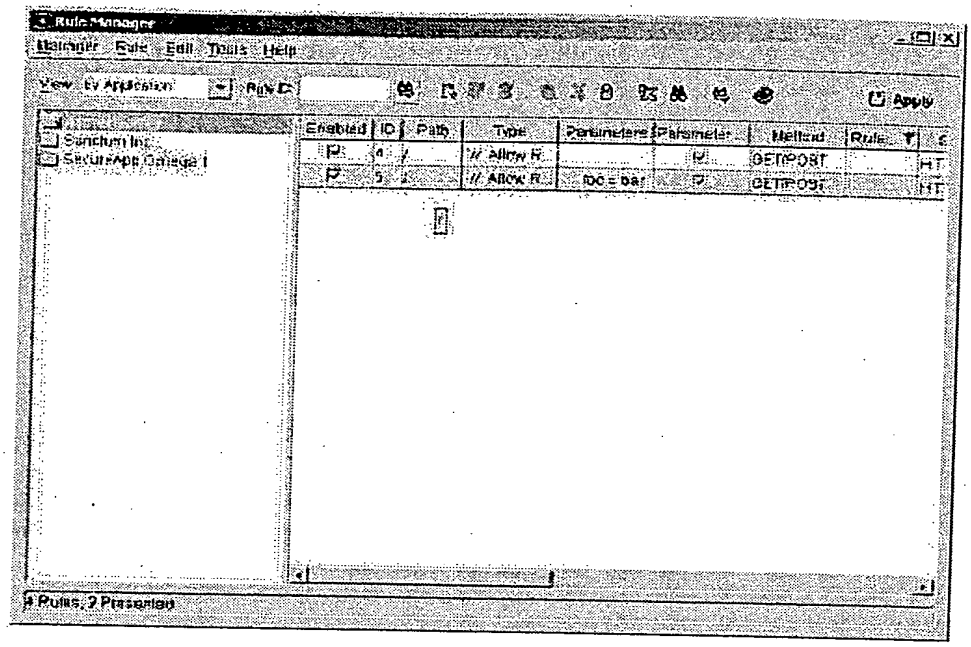

Figure 6

4.2 SSL Configuration

AppShield includes complete support for SSL enabled web sites and can be used as a central management point for any number of SSL applications and the SSL certificates for one or many web farms. Any number of SSL certificates can be installed into AppShield. You have the extra option of either passing the data un-encrypted on to the web server (freeing up web server resources) or re-encrypting the data stream for maximum security.

Installing and configuring SSL in AppShield is as simple as taking your existing SSL certificates already installed on your web servers and importing them into AppShield. In addition to this, you can also generate certificate requests within AppShield itself and submit the request directly to the certificate authority of your choice.

4.2.1 Client Side Certificates

AppShield 4.0 includes built in support for PKI enabled web applications that use SSL client certificates. Because AppShield is a proxy server, it breaks the connection, requiring that any SSL connections also be re-established. Client Side Certificates or CSCs present a unique problem for proxy severs since CSCs are designed to prevent interception. In order to support this method of authentication AppShield receives the SSL certificate, opens and analyzes its contents, and if valid, passes the individual client certificate data elements to the web application via special HTTP headers.

This allows the certificate data to pass to the back end web applications without being lost at the proxy. For companies that are looking to add PKI support in the future, AppShield provides the necessary infrastructure to start using PKI today even if your web applications are not yet PKI aware. With only a few changes, even legacy web applications can take advantage of the additional security offered by SSL Client Certificates.

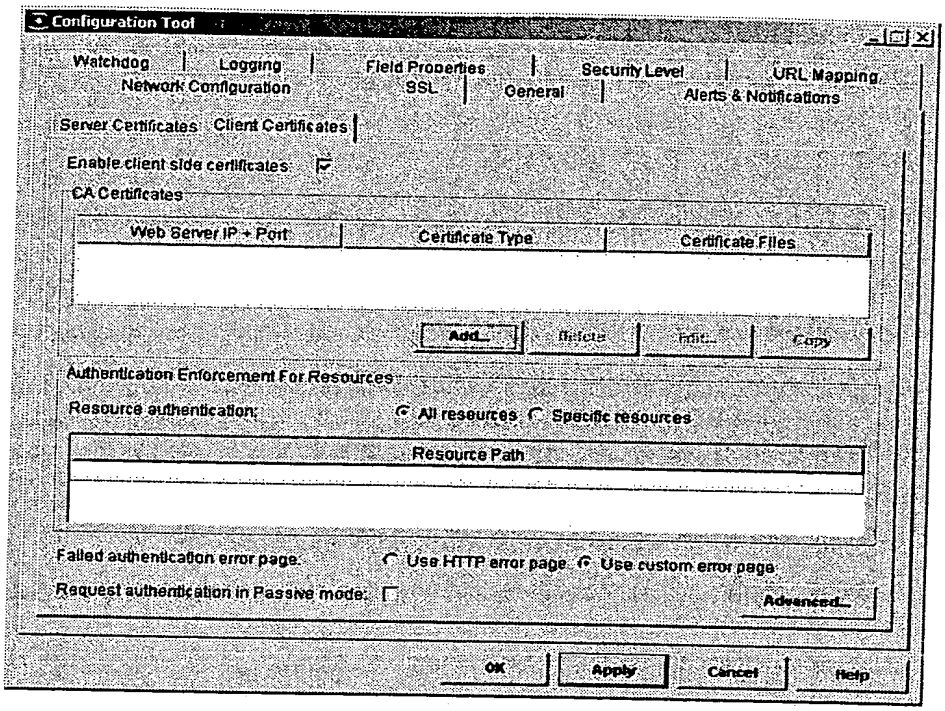

Figure 7

Below is an example of using the advanced option to specifically configure the HTTP header values that AppShield will pass to the web server.

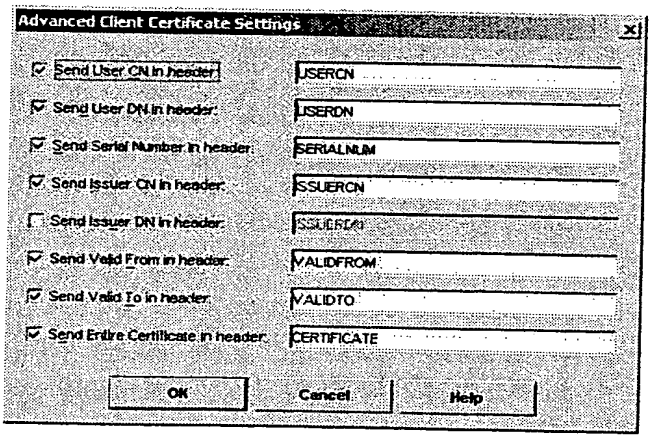

Figure 8

5 Management

AppShield's approach provides a powerful management system that makes the product very easy to use. All management is done via a Java-based GUI, which centrally controls all the AppShield units deployed in the site.

5.1 Management Console

The AppShield Management Console shown here provides a continuous, real-time readout of all web site activity, including pages-per-second, hits-per-second, and the total number of users online. In complex installations with multiple server locations, the Webmaster can manage all of them from a single console. To access each location, the Webmaster simply clicks the appropriate button at the bottom of the screen. In the example shown here, AppShield is tracking 120 hits/sec (10 million hits per day), 100 concurrent users, and one AppShield nodes as indicated by the buttons at the bottom of the screen. AppShield can also be managed remotely with multiple administrators able to view the console and one administrator who has read/write capabilities.

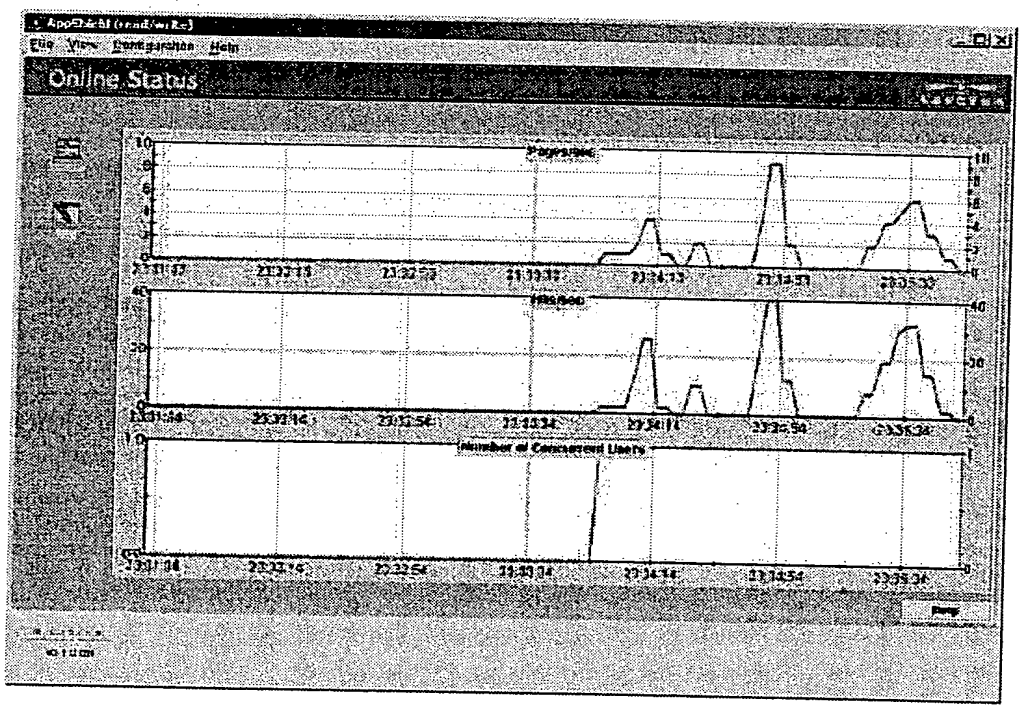

Figure 9

5.2 Attack Track™ - The AppShield logging system

AppShield gives users a complete, real-time log of all requests sent to the web server and system logs in Attack Track™. It displays valid, approved requests in black type and flags illegal requests in red type, with additional details in blue type for fast identification and enforcement action. Once identified, the Webmaster can quickly trace a hacking attempt back to its point of origin and perform analysis using the full details of the attack that is logged by AppShield

In addition, if the site uses SSL client certificates, the user name that is connected to the session will also be displayed and recorded by AppShield.

5.2.1 Privacy Compliance Controls

In addition to providing detailed information on the actions of your users, including valid and invalid requests, AppShield provides the administrator with the ability to hide sensitive fields from the logs. This makes sure that the confidential information of your customers such as credit card numbers and passwords is not recorded in the logs. To configure this option the administrator selects the "Hide Sensitive Data" option and configured the specific fields that should not be recorded in the logs. Instead of the actual data, a series of **** will be substituted and the actual values will not be stored anywhere within AppShield.

5.2.2 Log Management

AppShield logging uses a SQL database for storing and accessing the logged information that can easily be accessed by third party tools for log analysis. All events are time ordered with the viewer supporting resizable and positional fields with double-click expandability through pop-up windows. All recorded events can be used as a powerful forensics tool since they provide the time, source, and nature of the attack.

Alerts, information, and configuration suggestions are all displayed in the Attack Track window and can be clicked on for more information.

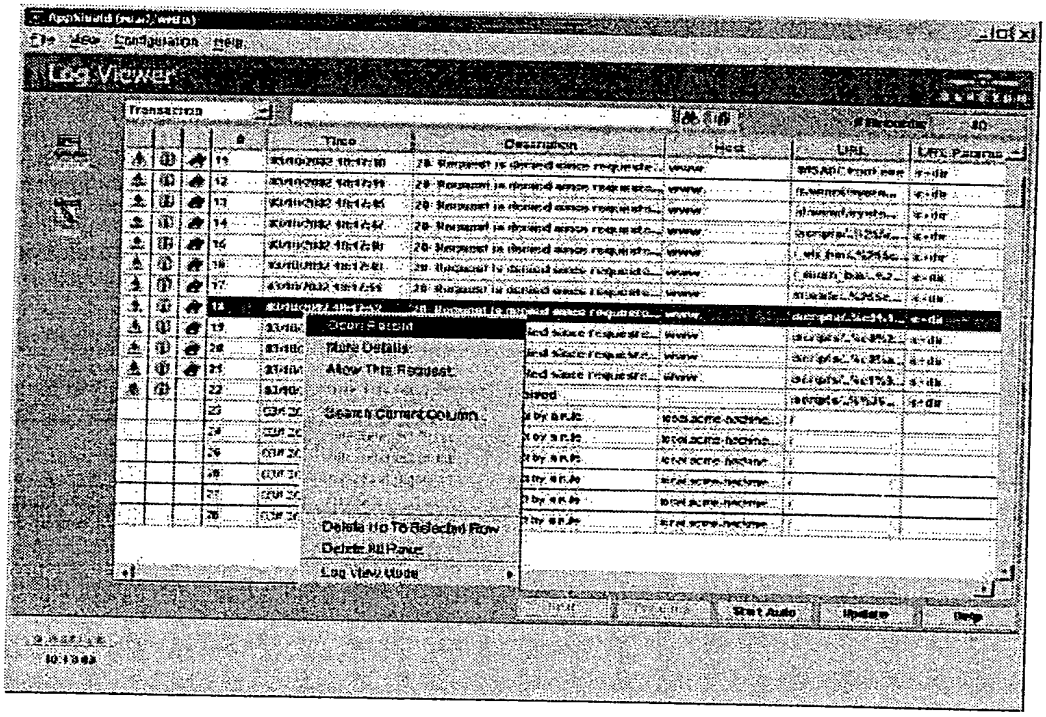

Figure 10

AppShield has a built in 'One Click' Rule Creation feature to assist in writing Policy Refinement rules. When right-clicking on a request that you want to allow you can choose "Allow this request" option that will create a rule to allow this request from now on.

5.2.3 Log Configuration and Fine Tuning

AppShield provides the ability for the admin to specify resource utilization and the type of image files to record or exclude from the log file, allowing for easy to maintain and easy to read logs. AppShield also allows you to backup your logs in a periodic manner so you can archive the full activity for as long as you require.

5.3 Alerting Features

Whenever AppShield encounters a hacking attempt it can issue a special alert. Supported alterting mechanisms are Popup console, SNMP, email, and OPSEC ready devices. Alert threashold settings are configurable through the management console.

5.4 AppShield Watchdog Technology

The watchdog technology is an independent component in AppShield that performs continuous monitoring of AppShield and the web server. It checks system parameters such as memory consumption, as well as the responsiveness of AppShield and the web server (using keep alive requests). If it detects an irregularity it can respond in a number of ways from alerting the operator to rebooting AppShield.

6 Deployment Options

6.1 Host Based Deployment

The Web Server version enables the deployment of AppShield without adding complexity to the network. It is installed on the same computer as the web server.

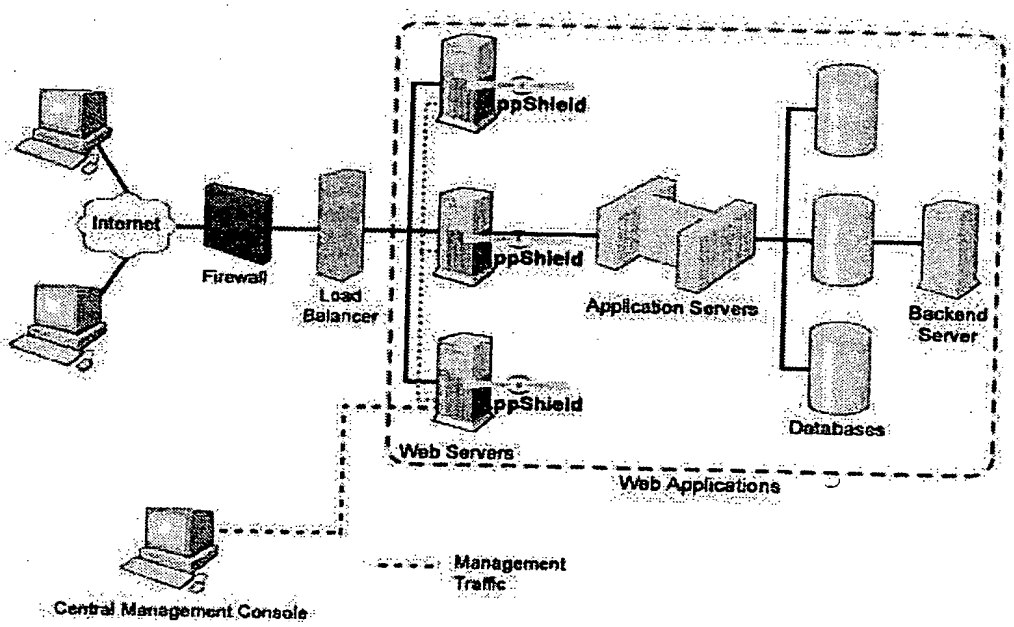

Figure 11

6.2 Gateway Deployment

Using AppShield as a gateway does not require any software installation on the web server system. In this configuration AppShield runs on a dedicated machine and connects to one or more web servers. The configuration below is fully compatible with existing load-balancer technology.

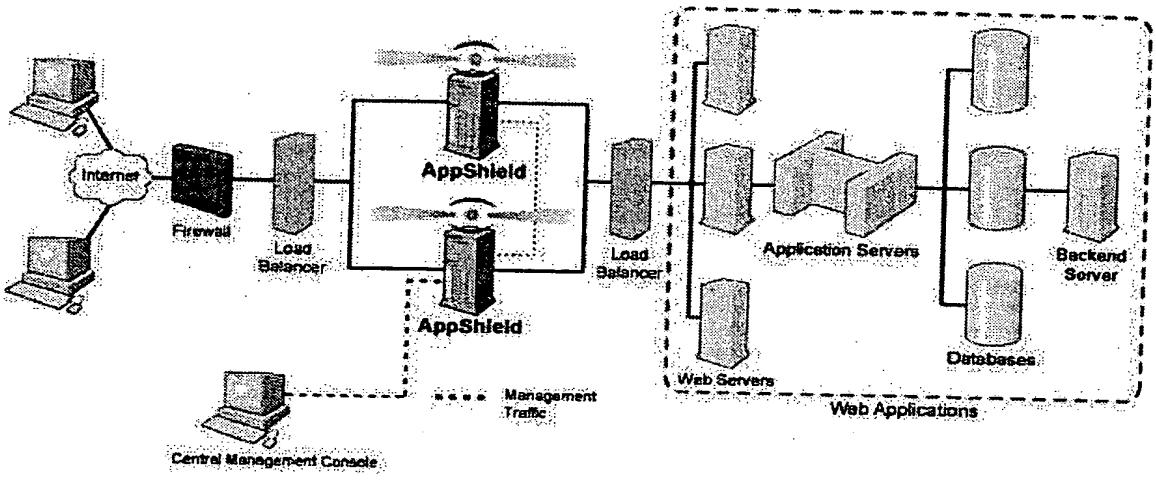

Figure 12

In either implementation, AppShield can run on Solaris (2.6, 2.7, and 8) or Windows (NT and 2000) platforms.

The Minimum System Requirements for the AppShield machine are:

Windows Minimum Requirements

- Computer: Pentium III PC, 500 MHz (Pentium III 933 MHz recommended)
- Operating System: Windows NT Server 4.0 with SP3 or Windows2000 SP2
- RAM: 512 Mbytes
- Network: 10/100 Mbps NIC (Dual 100Mbps NICs are recommended.)
- 16 bit color display (for management console)
- Disk: 500 MB free disk space

Solaris Minimum Requirements

- Computer: Sun Ultra II, 440 MHz
- Operating System: Solaris 2.6, 2.7 and Solaris 8 (Sparc only).
- In order to install AppShield, some patches available from Sun may be required. During installation, a list of missing patches will be generated. See "Appendix D-Solaris Patches," in the manual for a list of all recommended patches and information on how to check your system for the required patches.
- RAM: 512 Mbytes
- Network: 10/100 Mbps NIC (Dual 100Mbps NICs are recommended.)
- 16 bit color display (for management console)
- Disk: 500 MB free disk space?

7 Performance & Scalability

7.1 AppShield Performance

AppShield is designed to meet the high performance demands presented by growing e-Business sites. Its streamlined design combined with efficient SMP support enables AppShield to deliver high performance on a single CPU as well as scale to meet the needs of the largest and most demanding sites using SMP machines. The average added latency of AppShield is between 2-3 *milliseconds*. The graphs below show an analysis of transactions per second and throughput on a PIII 933MHz.

Test System: PIII 933 MHz, 100Mbps NIC

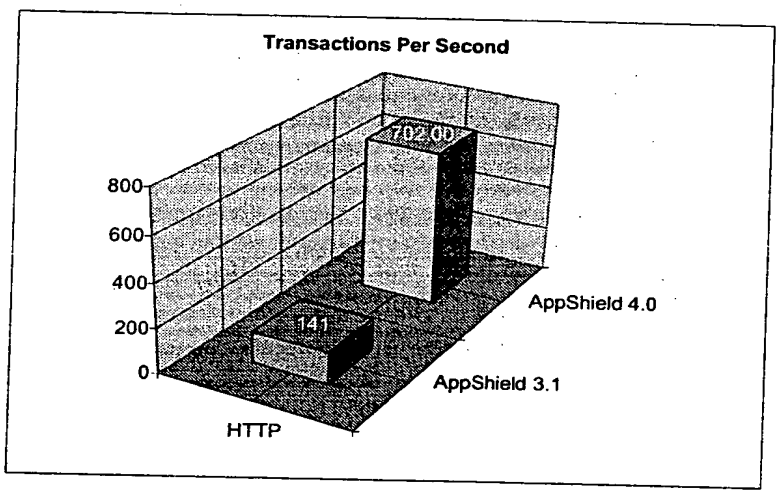

Figure 13

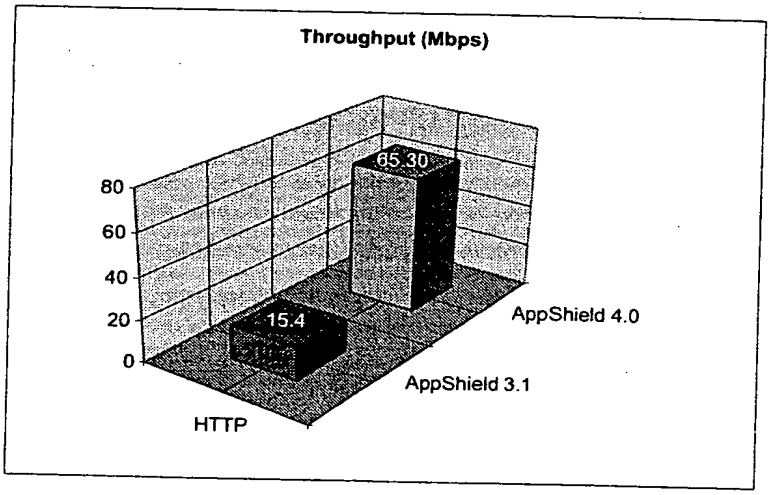

Figure 14

7.2 Scalability

AppShield is capable of supporting any number of concurrent users, limited only by the amount of memory used (the minimal recommended configuration allows for thousands of concurrent users). In addition, several AppShield nodes can be linked together, if needed, to protect a site or sites. In this configuration, a load balancer is used to distribute the load evenly across the AppShields. This can be the same load balancer that is also used by the web servers so additional hardware is not required. This enables the site owner to expand their site and add capacity by simply adding additional AppShields.

AppShield is also tuned to take advantage of multiple CPU systems effectively and will take advantage of upgraded hardware without modification.

7.2.1 Failover

As part of the scalability mechanism where there are multiple systems, there is also a built in method to handle any failure of the AppShield system. Each AppShield node maintains a link to the other nodes and can detect a node failure and immediately be ready to take on the traffic originally destined for the failed node. When the failed system comes back on line, it is automatically re-inserted into the AppShield farm.

8 Interoperability

AppShield is designed to work in the complex e-Business environment. AppShield interoperates with the following products and standards:

- **Any web server** such as iPlanet, Microsoft's IIS, and Apache
- **All major web browsers** such as Microsoft Internet Explorer, Netscape Navigator, Opera, and AOL Browser
- **All major load balancers** such as Cisco Systems Local Director, Radware's Web Server Director, F5's BIG/IP, and Resonate's Central Dispatch
- **Any application server** such as GemStone Systems, Secant Technologies, NetDynamics Application Server, Allair Cold Fusion, BroadVision and Oracle Application Server.
- **Any web statistics utility** capable of extracting user information from cookies such as WebTrends. Capable to create HTTP logs in a centralized place to ease statistics analysis.
- **Any firewall** such as Checkpoint's Firewall-1 and Cisco Systems' PIX Firewall
- **SSL Accelerators** such as Rainbow and nCipher
- SSL Global ID support
- SSL Client side certificates
- **SNMP** support for alerting
- **OPSEC** ELA (Event Logging API) certified
- **OPSEC** SAM supported
- **ODBC** support for exporting log file information

- **Internet Content support** for Cascading Style Sheets (CCS)
- **Heterogeneous Web site support** for shared hosting, multiple SSL formats, and multiple domain names
- Internationalized for double byte and most European languages

AppShield also works seamlessly with volatile source IP addresses (e.g., requests originated by AOL users).

9 Conclusion

AppShield provides a secured environment for applications through verification methods that assert that the application protocols are correct so that the application is used the way it was designed. AppShield ensures that users follow the application logic so applications protected with AppShield do not need to be built to cope with application hacking. Legacy applications are automatically protected with AppShield and do not need to be retrofitted. Secured applications can safely assume that: selections are always within a legal range (hyperlinks, form options, etc.); read-only, client-side data remains unmodified (hidden field, cookies, etc.); and free-format input is bound to be valid (text fields, password fields, etc). By using AppShield, security is implemented without impeding your web applications, effectively allowing your customers to access your applications to their fullest extent without allowing anyone to pervert them beyond their design scope and thus keeping your site safe and secured.

The integrity of an e-Business web site is the enterprise equivalent of national security. Companies need to do whatever possible to ensure the security of their sites and the digital assets they house. Businesses cannot wait passively for application-level security glitches to be discovered and fixed manually - it is a matter of brand, reputation and customer loyalty. Companies that address the application security problem with more appropriate solutions, such as Sanctum's AppShield, will find themselves in an enviable position in the crowded online marketplace. As the only ICSA certified Application Firewall, meeting stringent requirements to provide the best protection for your site, you can be assured that with AppShield your site is secure.

Sanctum Inc.
2901 Tasman Drive, Ste 205
Santa Clara, CA 95054
Phone: 877.888.3970 (US/Canada)
+972.9.958.6077 (Israel)

+44.7710.939512 (European HQ)
408.855.9500 (International)
Fax: 408.855.9521